Available online at www.sciencedirect.com

ScienceDirect

CAAI Transactions on Intelligence Technology 1 (2016) 72–78
http://www.journals.elsevier.com/caai-transactions-on-intelligence-technology/

Original article

Frame interpolation with pixel-level motion vector field and mesh based hole filling

Chuanxin Tang [a], Ronggang Wang [a], Zhu Li [b,*], Wenmin Wang [a], Wen Gao [a]

[a] *School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China*
[b] *School of Computing & Engineering, University of Missouri, Kansas City, MO 64110, USA*

Available online 4 June 2016

Abstract

Most of the traditional methods are based on block motion compensation tending to involve heavy blocking artifacts in the interpolated frames. In this paper, a new frame interpolation method with pixel-level motion vector field (MVF) is proposed. Our method consists of the following four steps: (i) applying the pixel-level motion vectors (MVs) estimated by optical flow algorithm to eliminate blocking artifacts (ii) motion post-processing and super-sampling anti-aliasing to solve the problems caused by pixel-level MVs (iii) robust warping method to address collisions and holes caused by occlusions (iv) a new holes filling method using triangular mesh (HFTM) to reduce the artifacts caused by holes. Experimental results show that the proposed method can effectively alleviate the holes and blocking artifacts in interpolated frames, and outperforms existing methods both in terms of objective and subjective performances, especially for sequences with complex motions.

Keywords: Frame interpolation; Motion estimation; Motion vector field; Frame rate up-conversion; Triangular mesh

1. Introduction

Frame rate up-conversion (FRUC) increases the number of displayed images per second in a video or film sequence. It is typically used in low bit rate video transmission. The source video is temporally down-sampled to a low frame rate and transmitted at a low bit rate. At the receiver, the original frame rate is recovered with FRUC. FRUC is also used to produce smooth motion or to convert video and film between different frame rates.

A simple FRUC method is frame repetition or frame averaging, which produces blurring and motion jerkiness around moving objects. To handle those problems, frames could be interpolated using *motion-compensated frame rate up-conversion* (MC-FRUC) method. Generally, MC-FRUC is composed of two steps: *motion estimation* (ME) and *motion-compensated frame interpolation* (MCFI). It creates new frames by first estimating motion trajectories between adjacent frames and then interpolating new frames along the motion trajectories. The quality of the created frames depends on the accuracy of the motion trajectories and the performance of the MCFI algorithm.

In conventional MC-FRUC algorithms, *block matching algorithm* (BMA) [1] is typically applied to estimate motion vectors and the new frame is interpolated along the motion trajectories. However, there are at least two problems in these methods. The first problem is the holes and collisions caused by occlusions and motion estimation errors. The second is the blocking artifacts in the interpolated frame caused by block-level motion vectors.

To handle the above mentioned problems, a number of algorithms have been proposed. For instance, bidirectional ME [2] and *overlapped block ME* (OBME) [3] are proposed to

* Corresponding author.
E-mail addresses: tangchuanxin1@163.com (C. Tang), rgwang@pkusz.edu.cn (R. Wang), lizhu@umkc.edu (Z. Li), wangwm@sz.pku.edu.cn (W. Wang), wgao@sz.pku.edu.cn (W. Gao).
Peer review under responsibility of Chongqing University of Technology.

http://dx.doi.org/10.1016/j.trit.2016.03.007

increase the accuracy of the estimated MVs. In order to improve the accuracy of the estimated MVs, several post-processing methods were proposed, including vector median filtering [4], adaptive vector median filtering [5], reliability-based MV processing [6], multistage MV processing [7], and correlation-based MV processing [8].

To handle collisions, the depth order of objects are determined and used in [9] and [10]. To fill in holes, a median filter is employed in [9], spatial interpolation is used in [10,11], and image inpainting is utilized in [12–14], block-wise directional hole interpolation is proposed in [15,16]. These algorithms cannot handle holes well especially when the hole areas are large.

To reduce blocking artifacts, the *overlapped block motion compensation* (OBMC) [1,2] was proposed. However, if a block is on the boundary of an object, blocking artifacts could still occur. Another approach for alleviating blocking artifacts is to use pixel-level *MV selection* (MVS) [15,16]. Besides, Dikbas and Altunbasak [17] proposed the pixel-based bilateral MVF from unidirectional MVs to enhance the motion accuracy. Although the above methods alleviate somewhat blocking artifacts, the motion vector candidates for each pixel are still obtained from *block motion estimation* (BME). Since pixels in the same block may belong to different objects, the MVs obtained from BME are not robust enough.

Our scheme to frame interpolation aims to overcome the problems presented previously. Optical flow algorithm is applied to estimate bidirectional pixel-level MVs. Motion postprocessing method based on image segmentation is utilized to improve the robustness of MVFs. The occluded regions are detected by motion trajectory tracking, and the detected occluded regions in the intermediate frame are generated by referencing either the previous frame or the next frame, and the non-occluded regions are generated by referencing both frames. At last, a new holes filling method using triangular mesh is proposed to handle holes. Our experiments testified the effectiveness of the proposed approach compared with conventional block-based methods.

The rest of the paper is organized as follows. The proposed method is presented in Section 2. Experimental results are discussed in Section 3. Finally, this paper is concluded in Section 4.

2. Proposed method

The proposed method uses two adjacent frames Ft and $Ft + 1$ to interpolate frame $Ft + 1/2$. Fig. 1 shows the overall block diagram of the proposed method. Bidirectional pixel-level MVs are estimated using optical flow algorithm in [18]. Then we upsample the reference frames by a factor of 2 both in horizontal and vertical directions and double the float MVs. Motion postprocessing based on image segmentation is proposed to improve motion spatial consistency. The intermediate frame is interpolated based on the post-processed MVFs. At last, remaining holes in the intermediate frame is handled and the intermediate frame is down-sampled to original size.

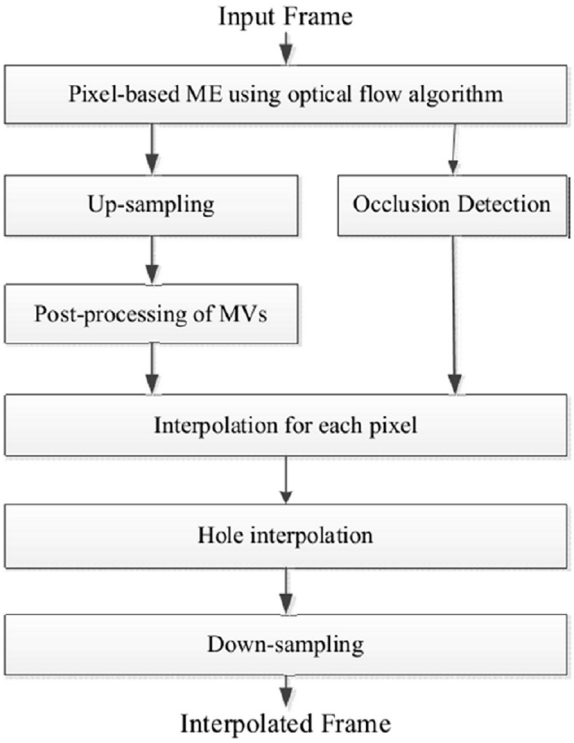

Fig. 1. The overall block diagram of the proposed system.

2.1. Bidirectional motion estimation

We use the optical flow algorithm proposed in [18] to estimate both forward and backward motion fields V_t^f and V_{t+1}^b between every two adjacent frames F_t and F_{t+1}. In the forward motion fields V_t^f, each motion vector is associated with a pixel in the previous frame and points to the next frame; whereas in the backward motion fields V_{t+1}^b, each motion vector is associated with a pixel in the next frame and points to the previous frame.

2.2. Up-sampling

The motion vector value obtained from optical flow is a float number. If it is round to integer, there will be jagged edges in the interpolated frame. Fig. 2(a) is an example of intermediate frame generated by integer MVs. To alleviate this problem, we up-sample the reference frames by a factor of 2 both in horizontal and vertical directions and double the float MVs before rounding. So the interpolated frame is also enlarged 2 times in both directions, we down-sample it to original size before outputting.

Fig. 2(b) shows the interpolated frame with the proposed method, from which we can see the artifact of jagged edges is mitigated.

2.3. Motion post-processing

On the other hand, since the MV of each pixel is estimated independently by optical flow. The MVs tend to be non-uniform even within the same object. In order to maintain

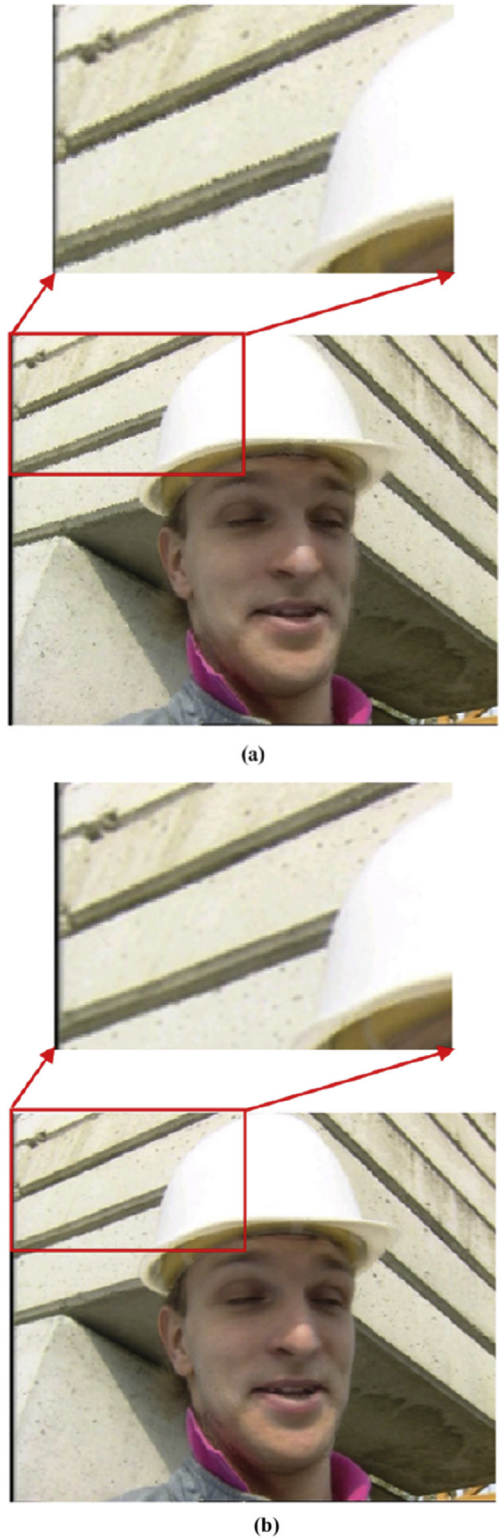

Fig. 2. (a) Without up-sampling and down-sampling (b) With up-sampling and down-sampling.

spatial consistency of pixel-level motion, post-processing of MVs based on image segmentation is utilized. For each MV, we select a window around the pixel and use the MVs within the window to smooth the MV of the center pixel. The window size is set to11 × 11. The updated MV is calculated as follows:

$$\mathrm{MV}c = \frac{1}{N}\sum_{j\in W}(\mathrm{MV}_j \cdot \alpha_j) \tag{1}$$

where W is the neighborhood pixel set around the current pixel and N is the number of pixel belongs to the same region with the center pixel within W and j is the index of pixel within the window and α is defined as

$$\alpha_j = \begin{cases} 1.\textit{if pixel j belongs to the same region?} \\ \textit{with center pixel} \\ 0.\textit{otherwise} \end{cases} \tag{2}$$

We use a simple image segmentation algorithm in [19] to segment the images into multiple regions.

2.4. Occlusion detection and MV refinement

Occlusion refers to the covered and uncovered areas only existing in one of the reference frames. The occluded regions in the intermediate frame should be generated from either the previous frame or the next frame. So it is important to detect occluded regions.

The reliability of MV is important for occlusion detection. The proposed method utilizes SAD and *MV Distance* (MVD) to determine whether the MV is reliable. SAD is the absolute difference between corresponding pixels and MVD is determined as follows:

$$MVD_t(x) = \left\| V^f_{t,x} - V^b_{t+1,x'} \right\|, for\ x + V^f_{t,x} = x' \tag{3}$$

where $\|\bullet\|$ is the Euclidean distance. If the SAD of a pixel is smaller than a threshold of Th_SAD and the MVD described in Eq. (3) is smaller than a threshold of Th_MVD, then the MV of the pixel is reliable.

The occluded regions are detected by tracking motion trajectory as shown in Fig. 3. In Fig. 3, there are three consecutive frames and the pixel $P1$ in frame t is occluded by

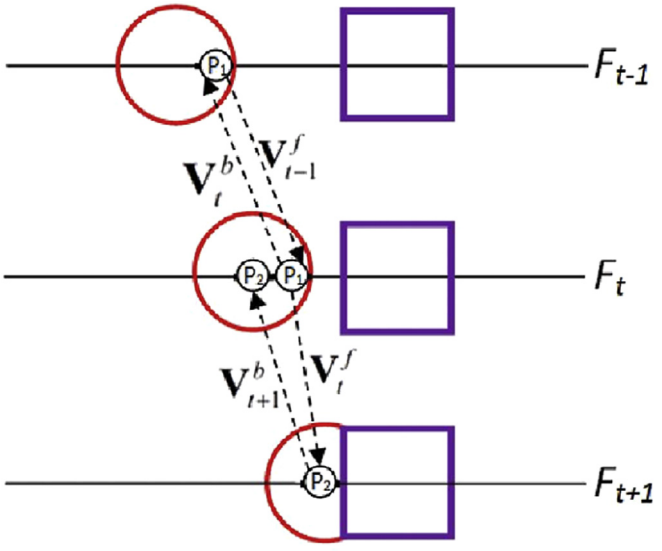

Fig. 3. Occlusion Detection based on the motion trajectory.

the rectangle in frame t + 1. So we can obtain correct MV(V_t^b) of P1 between frame t − 1 and t, but fail to get correct MV(V_t^f) between frame t and t + 1. The correctness of the MV is determined by the mentioned SAD and MVD. Then, we mark the V_t^f of P1 as occlusion and set the value of V_t^f as $-V_t^b$.The reason we use opposite direction of V_t^b to update V_t^f is that we suppose pixels move continuously among successive frames.

2.5. Interpolation

Interpolation stage consists of two steps: first, using both forward and backward MVFs to generate the forward interpolated frame F^f and the backward interpolated frame F^b; second, merging F^f and F^b to generate the final interpolated frame $F_{t\ +\ 1/2}$.

Step 1: Generating F^f and F^b is the overall block diagram of computing F^b. For each pixel to be warped, if its MV is not reliable and is not marked as occlusion, then drop the pixel and deal with the next pixel. If its MV is reliable or marked as occlusion, then determine whether collisions occur. If pixel P and a previous pixel P' are warped to the same position, then it is defined as 'collisions'. If collisions occur and current MV of pixel P is less reliable than the MV of the previous pixel P', then drop the pixel and deal with the next pixel. Otherwise, we warp the pixel to the Ff. The forward warping equation is defined as follows,

$$F^f\left(x+\frac{V_t^f(x)}{2}\right)=F_t(x) \tag{4}$$

where x is the coordinate of the pixel in frame t.

The method of computing F^b is the same as F^f described in Fig. 4. The backward warping equation is defined as follows,

$$F^b\left(x+\frac{V_{t+1}^b(x)}{2}\right)=F_{t+1}(x) \tag{5}$$

where x is the coordinate of the pixel in frame t + 1.

1) Step 2: Merging F^f and F^b

The interpolated frame of $F_{t\ +\ 1/2}$ is obtained by the following process,

$$F_{t+1/2}(x,y)\begin{cases}\dfrac{F^f(x,y)+F^b(x,y)}{2}, if\ F^{'}(x,y)\neq Hole\\ F^f(x,y), if\ F^f(x,y)\neq Hole\ and\ F^b(x,y)=Hole\\ F^b(x,y), if\ F^f(x,y)=Hole\ and\ F^b(x,y)\neq Hole\\ Hole, otherwise\end{cases} \tag{6}$$

where **x** and **y** represent the pixel location. If both the forward interpolated frame F^f and the backward interpolated frame F^b have different values, the value of the interpolated pixel is set as the averaging of the two pixel values. If only one value is

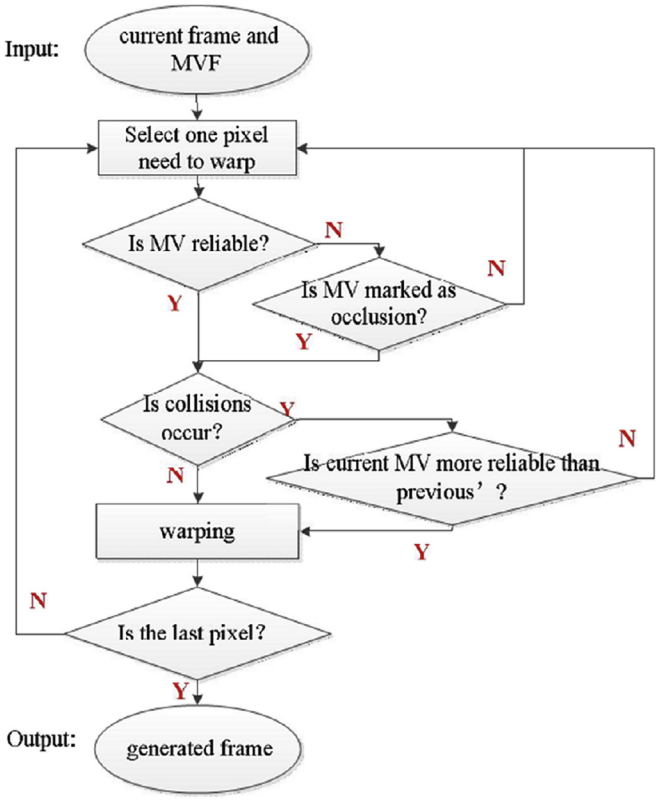

Fig. 4. Block diagram of generating forward interpolated frame (F^f).

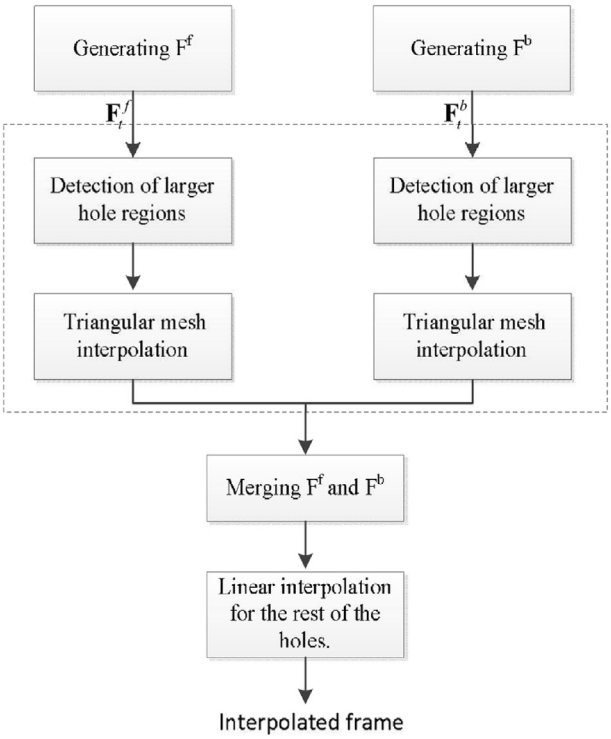

Fig. 5. Block diagram of the HFTM.

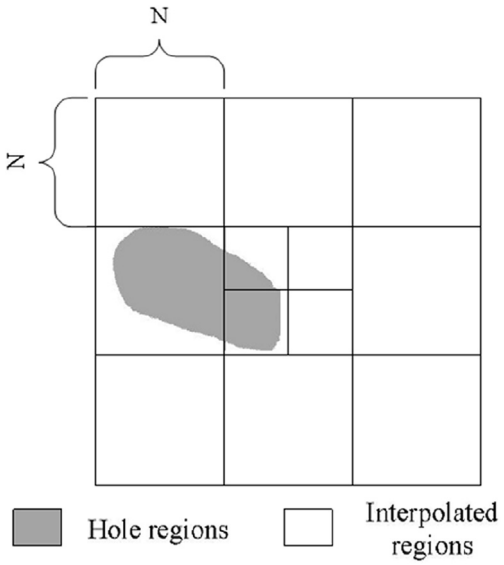

Fig. 6. Illustration of the detection of larger hole regions.

available since there is a hole in either F^f or F^b, the value of to be interpolated pixel is set as this available value. Otherwise, the pixel is a hole.

2.6. Holes filling using triangular mesh

As is shown in Fig. 5, the proposed HFTM consists of three steps: first, the detection of larger hole regions; second, triangular mesh interpolation; third, linear interpolation for the rest of the holes.

1) Step 1: Detection of larger hole regions

As shown in Fig. 6, the interpolated frame is split into square sub-blocks of size $N \times N$. If the number of pixels belonging to holes in each sub-block is larger than a threshold of Th_Hole, the hole pixels in the sub-block is marked as larger hole regions.

Then, the interpolated frame is recursively split into square subblocks of size $N/2 \times N/2$. If the number of hole pixels in each sub-block is larger than a threshold of Th_Hole/4, the hole pixels in the sub-block is marked as larger hole regions.

2) Step 2: Triangular mesh interpolation

The interpolated frame is split into blocks of size $N \times N$. For each block, the pixel with the smallest MVD is selected as the triangular mesh control points. A triangle is formed from the triangular mesh control points from the neighboring blocks, as shown in Fig. 7.

The motion within the triangle is assumed to be affine. Therefore, a pixel within the triangle in frame t is mapped to a pixel in frame t' by the following relationship,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_0 \\ a_4 & a_5 & a_3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{7}$$

The above procedure is only applied to the pixels that belong to the larger hole regions.

3) Step 3: Linear interpolation for the rest of the holes.

After merging F^f and F^b to generate $F_{t+1/2}$, there are few holes. For the rest of the holes, linear interpolation from the four nearest neighbors is employed.

3. Experimental results and discussion

The performance of the proposed frame up-sampling algorithms are evaluated by several experiments. The CIF sequences of Football, Foreman, Bus, Ice, Highway, and Soccer are used. Even frames are skipped and the frame interpolation methods are applied to interpolate the skipped frames. All the test sequences have a frame rate of 30 frames per second and each of them consists of 101 frames.

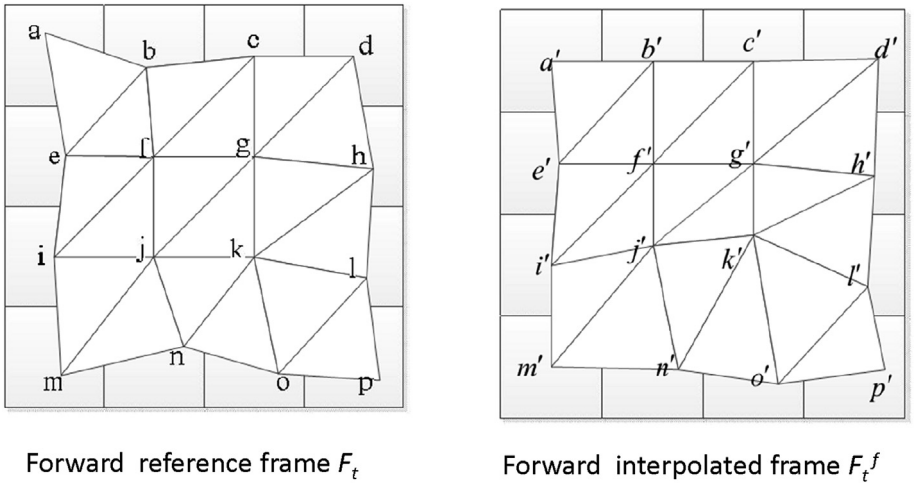

Fig. 7. Illustration of triangular mesh interpolation.

The proposed method is compared with the recent state of the art approach in [17] and other two block-based MCFI methods: dual-ME [20], correlation [21] methods and our last work in [22]. In the proposed method, Th_MVD is set to 3, Th_SAD is set to 20, Th_Hole is set to N × N × 15/16 and N is set to 16.

3.1. Comparison of the objective performance

The objective evaluation is based on PSNR. Table 1 presents the average PSNRs of 50 interpolated frames by various methods. From the table, we can see that our last work in [17] and the proposed method obtain significantly improved objective results compared to the other MCFI methods. Our last work in [22] outperforms the other methods for all the test sequences except for one sequence and obtains better results than the algorithm in [17]. The average improvement in PSNR is 1.02 dB.The proposed method in this paper has almost the same objective performance as the algorithm in [22], and the average loss in PSNR is 0.04 dB. It is clear that proposed method has quite some PSNR quality gains over other group's work in [17,20,21].

3.2. Comparison of the subjective quality

The interpolated frames by MCFI methods have poor subjective quality mainly due to blocking artifacts and occlusions. The algorithm in [17,22] and the proposed method do not suffer from the blocking artifacts. So the proposed method is just compared to the algorithm in [17] and [22] for subjective quality.

Figs. 8 and 9 shows the original images of *Foreman* and *Bus* together with the interpolated results of the considered three methods. As is shown in Fig. 8, the proposed method and the method in [22] obtain better subjective visual quality compared with the algorithm in [17]. Since there is on larger hole region in the interpolated frame, the proposed method and the method in [22] produce same performance in visual quality. However, on a sequence with more challenging motion, such as *Foreman*, the difference among the interpolated frames becomes more obvious, as is shown in Fig. 9. Interpolated frame in Fig. 9(b) creates obvious artifacts in neck. Interpolated frame in Fig. 9(c) still produces blurring in neck. The frame produced by the proposed method in Fig. 9(d) has the overall highest quality. As mentioned before, most of blocking artifacts are relieved in the proposed FRUC; however, some artifacts still exist in a few

Table 1
Average Psnrs of the 50 Interpolated Frames from Various Methods.

Video Sequences	Dual ME [20]	Correlation [21]	[17]	Without HFTM [22]	Proposed
Football	22.38	22.63	22.58	**23.34**	23.23
Foreman	33.24	33.46	34.31	34.96	34.93
Bus	25.53	25.24	26.86	28.55	28.65
Ice	29.30	31.74	32.24	34.36	34.18
Highway	–	–	33.23	33.51	33.49
Soccer	–	–	29.33	30.85	30.81
Average	–	–	29.76	30.93	30.89

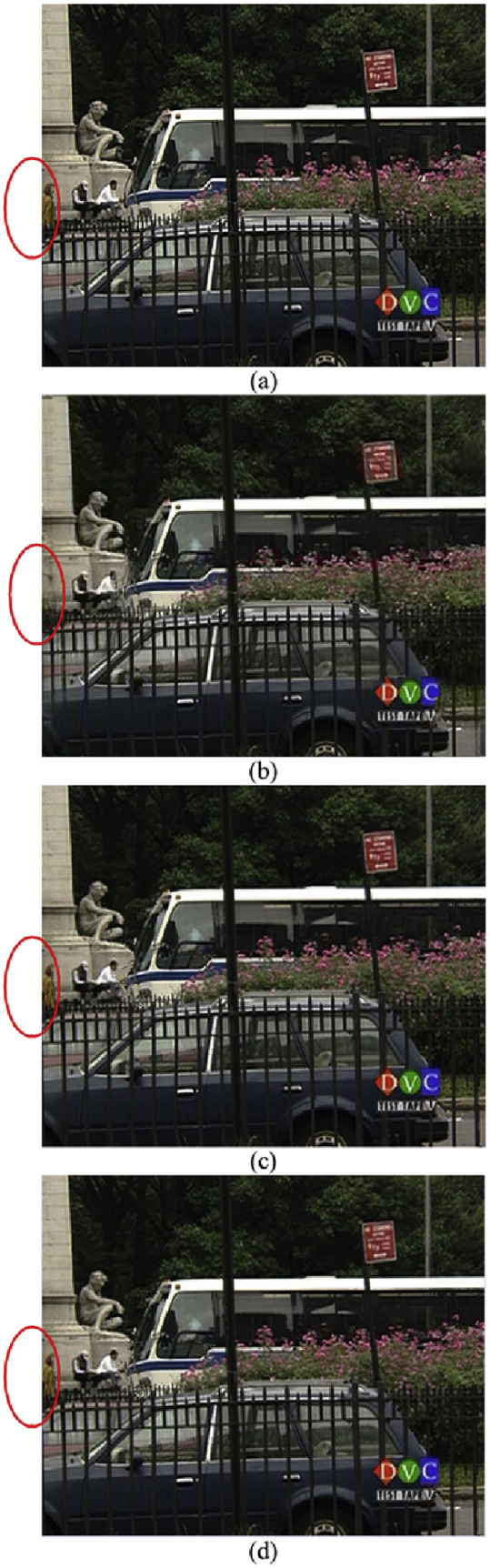

Fig. 8. Results for images *Bus* frame 12. (a) Original image. (b) Output of [17]. (c) Output of [22]. (d) Output of the proposed method.

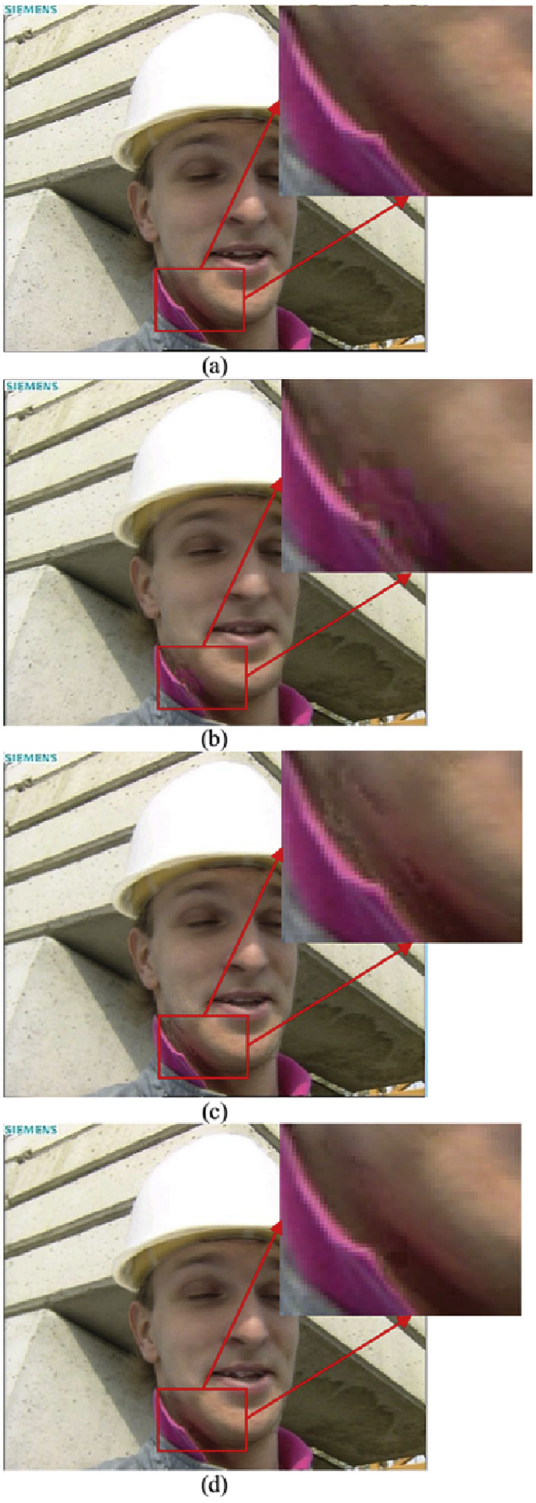

Fig. 9. Results for images *Foreman* frame 58. (a) Original image. (b) Out-put of [17]. (c) Output of [22]. (d) Output of the proposed method.

sequences with various motion activity. Comparing the original frame of *Foreman* sequence, the interpolated frames of the conventional methods and the proposed method show some artifacts in neck. However, the proposed method can yield better quality than others because of its way of dealing hole regions. The hole regions are handled with the proposed holes filling method.

4. Conclusion

In this paper, a new frame interpolation method with pixellevel MVF is presented. The proposed method utilizes bidirectional pixel-level MVFs obtained by optical flow algorithm to alleviate blocking artifacts. Motion post-processing is proposed to keep spatial consistency. At the interpolation stage, a new warping method considering occlusions is proposed to gracefully obtain the interpolated frame. At last, a new holes filling method using triangular mesh is proposed. A good performance gain was achieved by the proposed method compared with traditional methods, especially for sequences with fast motions.

References

[1] K. Hilman, H. Park, Y. Kim, IEEE Trans. Circuits Syst. Video Technol. 10 (6) (2000) 869–877.
[2] B.D. Choi, et al., IEEE Trans. Circuits Syst. Video Technol. 17 (4) (2007) 407–416.
[3] T. Ha, S. Lee, J. Kim, IEEE Trans. Consumer Electron. 50 (2) (2004) 752–759.
[4] J. Astola, P. Haavisto, Y. Neuvo, Proc. IEEE 78 (4) (Apr. 1990) 678–689.
[5] L. Alparone, M. Barni, F. Bartolini, V. Cappellini, Adaptively weighted vector-median filters for motion-field smoothing, in: Proc. Int. Conf. Acoust. Speech Signal Process, vol. 4, May 1996, pp. 2267–2270.
[6] Y.T. Yang, Y.S. Tung, J.L. Wu, IEEE Trans. Circuits Syst. Video Technol. 17 (12) (Dec. 2007) 1700–1713.
[7] A.M. Huang, T.Q. Nguyen, IEEE Trans. Image Process. 17 (5) (May 2008) 694–708.
[8] A.M. Huang, T.Q. Nguyen, IEEE Trans. Image Process. 18 (4) (Apr. 2009) 740–752.
[9] J. Benois-Pineau, H. Nicolas, J. Vis. Commun. Image Represent. 13 (2002) 363–385.
[10] P. Blanchfield, D. Wang, A. Vincent, F. Speranza, R. Renaud, SMPTE Motion Imaging J. (Apr. 2006) 153–159.
[11] A. Kaup, T. Aach, Efficient prediction of uncovered background in interframe coding using spatial extraprediction, in: Proc. ICASSP, Apr. 1994, pp. 501–504.
[12] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, "Image In Painting," in Computer Graphics (SIGGRAPG 2000), Jul. 2000, pp. 417–424.
[13] S.D. Rane, G. Sapiro, M. Bertalmio, IEEE Trans. Image Process. 12 (3) (Mar. 2003) 296–303.
[14] A. Criminisi, P. Perez, K. Toyama, IEEE Trans. Image Process. 13 (9) (Sep. 2004) 1200–1212.
[15] D. Wang, et al., IEEE Trans. Broadcast. 56 (2) (2010) 142–149.
[16] T.H. Tran, C.T. LeDinh, IEEE J. Sel. Top. Signal Process. 5 (2) (2011) 252–261.
[17] S. Dikbas, Y. Altunbasak, IEEE Trans. Image Process. 22 (8) (2013) 2931–2945.
[18] L. Xu, J. Jia, Y. Matsushita, Motion detail preserving optical flow estimation, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, pp. 1744–1757.
[19] R. Adams, L. Bischof, IEEE Trans. Pattern Analysis Mach. Intell. 16 (6) (1994) 641–647.
[20] S.J. Kang, S. Yoo, Y.H. Kim, IEEE Trans. Circuits Syst. Video Technol. 20 (12) (2010) 1909–1914.
[21] A.M. Huang, T. Nguyen, IEEE Trans. Image Process. 18 (4) (2009) 740–752.
[22] Chuanxin Tang, Ronggang Wang, Wenmin Wang, et al., A new frame interpolation method with pixel-level motion vector field, in: Visual Communications and Image Processing, 2014 IEEE International Conference on, IEEE, 2014, pp. 350–353.